Increasing Undergraduate Research in Computational Sciences

Maria Pantoja
Computer Science
California Polytechnic State University
San Luis Obispo, USA
mpanto01@calpoly.edu

Chris Lupo
Computer Science
California Polytechnic State University
San Luis Obispo, USA
clupo@calpoly.edu

***Abstract*—High Performance Computing (HPC) is used everywhere from e-commerce to machine learning. The main problems solved by HPC are related to computational sciences and affect almost all scientific disciplines. Training the students that will have sufficient knowledge in HPC is a crucial task of computer science departments in all educational institutions. This paper describes the steps taken to increase the undergraduate engagement in HPC disciplines in our undergraduate institution. The increased exposure allowed our computer science students to collaborate with other departments research projects by accelerating different computational intensive problems. In order to increase students knowledge in HPC we decided to offer classes at the undergraduate level in parallel programming and distributed systems; and to properly support these classes we invested in a lab specifically designed for parallel programming, with machines equipped with the best multicores cpus and gpus so students can experience the effects of massively parallel architectures in their code. Using detailed enrollment, master thesis and publications from the last few years, since the last has been available; we can see a substantial increase in undergraduate students interest in the HPC area, and a great success in the number of research master thesis as well as the number of peer reviewed undergraduate and graduate student-authored publications.**

***Index Terms*—HPC, Undergraduate Research, CUDA, Distributing Computing, Parallel Programming**

I. Introduction

Computational approach to solve scientific problems require interdisciplinary skills. Scientific instruments, from particle accelerators to DNA sequencers, and consumer electronic devices, from thermostat to security cameras, are generating data faster than our capacity to analyze it. Modern computational paradigms such as multicore CPU, many-core architectures like GPUs, and cloud computing, are making HPC hardware more affordable. What we are still lacking is specialists that can efficiently program this systems. In this paper we present details of the steps taken at our undergraduate institution to increase the exposure of our students to HPC disciplines and the successes we had with them.

In this paper we present our experience as an undergraduate institution after dedicating a lab exclusively for parallel and distributed computing, which mainly targets undergraduate students in computer science related disciplines, but is also open to other schools and colleges since the only pre-requisite to take the class are data structures and one class in C programming or equivalent.

Partially inspired by the NSF/TCPP curriculum initiative on Parallel and Distributed Computing, [1], and our relations with the industry, around 2012 our institution started the fundraiser for a lab specifically for massively parallel applications. The goal was to create interest in HPC for our students and this way improve their collaborative research experience. To achieve the goal we introduced two distinctive classes in HPC, Parallel Programming and Distributed Computing with the intention of providing students with a lab that has the latest hardware accelerators so they can practice and measure the acquired skills.

To improve the number of students taking HPC classes, prerequisites are kept at a minimal so biology, chemistry, physics , mathematics and other sciences can take the class. We are aware that the majority of the students taking these classes are going to be computer scientist or related disciplines or students doing at least a computer science minor; but we encourage collaboration and multidisciplinary environments in the capstone projects where students abilities can be divided in computational skills and science skills to work together in a common project that needs acceleration.

A. Class Description

There are two main classes added as electives to undergraduates. Our institution is on the quarter system so all classes are 10 weeks long.

1) Parallel Programming. The programming languages used for the class are: Pthreads, OpenMP, and CUDA. The main topics and assignments are listed in table I
2) Distributed Computing. The programming languages used for the class are MPI, and GoLang. The main topics and assignments are listed in table II.

Since both classes have been offered they have been always filled at room capacity. We do keep our classes relatively small in size with a capacity of around 30 students per class.

II. Results

To evaluate the success of the new lab and classes we provide a partial list of student lead research (list II). For us is very important to use the amount of research produced as a metric of success since we know that increase in research experience

TABLE I
PARALLEL PROGRAMMING CONTENT

Topic	Assignment
Why Par. Prog. ; Intro to Multicore Programming	Matrix Multiplication in Pthreads, with timing
OpenMP Intro. (Parallel for, Reductions and Scheduling of Imbalanced Loops	Matrix Multiplication and Convolution in OpenMP, with timing
OpenMP Adv. Task Parallelism,SIMD, Vectorization, target Offload	Diffusion Simulation in OpenMP
GPU Prog. CUDA. GPU kernels. Thread hierarchy. Refactor serial loops. Allocate and free memory in both CPUs and GPUs. Handle errors generated by CUDA code.	vector addition in CUDA, with timing
Profiling GPU code. Streaming Multiprocessors to optimize execution configurations. Unified Memory. CUDA optimization.	matrix mult and conv. code in CUDA, with timing and comparison with above
Atomics, Advance Shared Memory and bank conflicts optimization	N-Body simulation, Histogram Equalization on an Image

TABLE II
DISTRIBUTED SYTEMS CONTENT

Topic	Assignment
Why Distributed Systems. Go Tutorial	String Search in GoLang
Process review. Communication. RPC, Message oriented, Multicast(Flooding,Gossip)	Gossip Heartbeat protocol in GoLang
Coordination. Clock Mutual exclusion	Written exercise on timing and M.E. for distributed databases
Coordination. Election algo. Gossip based coordination	Simple Paxos in GoLang.
Distributed Hash Tables	Dynamo Ring hash table implementation
MPI tutorial.	MM and convolution in MPI
Sorting on a Distributed System. Overlapping memory with calculations	Sorting in MPI
Fault Tolerance	Simple Fault detection in GoLang or MPI

has been identified as a highly beneficial for increasing student engagement and retention in STEM, specifically is been shown to be effective on attracting underrepresented minority students. In the last 4 years students leaded publications on HPC average 10 conference and journals papers per year. An incomplete but substantial list of the projects researched by our students is as follows.

1) Lazy Fault Recovery For Redundant MPI. A recovery mechanism based on replicas that works on top of the asynchronous fault detection implemented in previous work. Results shows that our recovery implementation is successful and the overhead in execution time is minimal. In Proceedings of the 6th Latin American Conference of High Performance Computing, CARLA 2019. Springer International Publishing, 2019.
2) Acceleration of Radio Frequency Propagation with General Purpose GPU Computing. CMMSE Conference 2019.
3) "Optimizing the Distributed Hydrology Soil egetation Model For Uncertainty Assessment with Serial, Multicore and Distributed Accelerations". In Proceedings of the 5th Latin American Conference of High Performance Computing, CARLA 2018. Springer International Publishing, 2018.
4) GPU Acceleration for Directional Variance based Intra prediction in HEVC. In Proceedings of the 5th Latin American Conference of High Performance Computing, CARLA 2018. Springer International Publishing, 2018.
5) Distributed Execution of Communicating Sequential Process-Style Concurrency. Journal of SuperComputing. Springer, 2018
6) A Heterogeneous Compute Solution for Optimized Genomic Selection. In Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine. IEEE, 2014.
7) GPUMap: A transparently GPU-accelerated Python map function. In Proceedings of the 7th Workshop on Python for High-Performance and Scientific Computing, PyHPC 17, New York, NY, USA, 2017.
8) TARUC: A topology-aware resource usability and contention benchmark. In Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering, ICPE 17, New York, NY, USA, 2017. ACM.
9) Enhancing regional ocean modeling simulation performance with the Xeon Phi architecture. In OCEANS 2017 - Aberdeen, June 2017.

Increase of papers published by students by a factor of X Master thesis in collaboration with other departments: Environmental Science Agriculture Chemistry The interdisciplinary nature of HPC clearly shows in the diversity of project selected by our students.

W

REFERENCES

[1] Chtchelkanova A. Dehne F. Gouda M. Gupta A. Jaja J. Kant K. La Salle A. LeBlanc R. Lumsdaine A. Padua D. Parashar M. Prasanna V. Robert Y. Rosenberg A. Sahni S. Shirazi B. Sussman A. Weems C. Prasad, S. K. and J. Wu. Nsf/ieee-tcpp curriculum initiative on parallel and distributed computing - core topics for undergraduates, version i. *NSF*, 0(0):10–16, 2012.